

## Claims

- [c1] 1. A pipelined incrementer comprising:
- count registers for storing a current count of the pipelined incrementer, the count registers receiving sum bits that are stored as the current count in response to a clock input;
  - pre-carry registers, receiving pre-carry signals that are stored as pipelined carry lookahead signals in response to the clock input;
  - pre-carry logic, receiving at least some of the sum bits, for generating the pre-carry signals as a function of a next count indicated by the sum bits, the next count being after the current count in a pre-determined sequence; and
  - sum logic, receiving the current count from the count registers, and receiving the pipelined carry lookahead signals from the pre-carry registers, for generating the sum bits that indicate the next count, whereby carry signals are generated the sum bits that indicate the next count, stored in the pre-carry register, and used by the sum logic in a following clock cycle.
- [c2] 2. The pipelined incrementer of claim 1 wherein the sum logic further comprises for at least some bit-positions of the current count:
- toggle logic, receiving a count bit for a bit-position of the current count from one of the current count registers, and receiving a local carry-in signal for the bit-position, the toggle logic toggling the count bit to generate the sum bit for the bit-position in response to the local carry-in signal for the bit-position; and
  - all-ones detect logic, receiving count bits for lesser bit-positions from some of the current count registers, and some of the bit-positions of the current count receiving the pipelined carry lookahead signals instead of some of count bits for lesser-significant bit-positions of the current count, for activating the local carry-in signal when all lesser-significant bit-positions are logical one, whereby a count bit for a bit-position is toggled when all lesser-significant bit-positions of the current count are logical one as indicated by the count bits or the pipelined carry lookahead signals.
- [c3] 3. The pipelined incrementer of claim 2 wherein the toggle logic is replicated for all bit-positions of the current count above a least-significant bit-position;

wherein the all-ones detect logic is replicated for bit-positions of the current count above two least-significant bit-positions,  
whereby toggle and all-ones detect logic is replicated for several bit-positions.

- [c4] 4.The pipelined incrementer of claim 2 wherein the all-ones detect logic is not replicated for an upper bit-position of the current count that corresponds to a first pipelined carry lookahead signal that is generated from all bit-positions below the upper bit-position.
- [c5] 5.The pipelined incrementer of claim 2 wherein the toggle logic for a bit-position comprises an exclusive-OR (XOR) gate or an exclusive-NOR (XNOR) gate.
- [c6] 6.The pipelined incrementer of claim 2 wherein the all-ones detect logic for a bit-position comprises an AND gate or a NAND gate that receives a group of inputs selected from the count bits and the pipelined carry lookahead signals.
- [c7] 7.The pipelined incrementer of claim 6 wherein the pre-carry logic comprises a first AND or NAND gate that receives as inputs a first group of lowest-significance bit-positions of the sum bits, and a second AND or NAND gate that receives as inputs a second group of next-lowest-significance bit-positions of the sum bits that are in more significant bit-positions than the first group.
- [c8] 8.The pipelined incrementer of claim 7 wherein the sum bits comprise at least 7 bits and the count bits comprise at least 7 bits and the pipelined carry lookahead signals comprise at least 2 signals,  
whereby the pipelined incrementer is at least a 7-bit counter.
- [c9] 9.The pipelined incrementer of claim 8 wherein the next count follows the current count in a binary-number sequence.
- [c10] 10.The pipelined incrementer of claim 2 further comprising:  
reset logic, coupled to the count registers and to the pre-carry registers, for resetting the current count to an initial value in response to a reset signal.
- [c11] 11.The pipelined incrementer of claim 10 wherein the reset logic comprises NOR gates between the sum bits and inputs to the count registers, the NOR

gates receiving the reset signal,  
whereby reset is synchronous to the clock input.

[c12]

12.A sequencer comprising:

sequence register means, responsive to a clock, for storing sum bits input to the sequence register means as current state bits in response to the clock;  
first carry register means, responsive to the clock, for storing a first pre-carry signal that is input to the first carry register means, and for outputting a first pipelined carry signal;  
first pre-carry logic means, receiving a first group of the sum bits, for generating the first pre-carry signal;  
second carry register means, responsive to the clock, for storing a second pre-carry signal that is input to the second carry register means, and for outputting a second pipelined carry signal;  
second pre-carry logic means, receiving a second group of the sum bits, for generating the second pre-carry signal; and  
combinatorial logic means, receiving the current state bits from the sequence register means and receiving the first and second pipelined carry signals, for generating the sum bits,  
whereby pre-carry signals are generated from the sum bits and stored for use in a next cycle of the clock.

[c13]

13.The sequencer of claim 12 wherein the combinatorial logic means further comprises:

toggle means for each sum bit, each toggle means receiving a current state bit and generating a sum bit for a different bit-position, the toggle means for inverting a logical state of the current state bit to generate the sum bit.

[c14]

14.The sequencer of claim 13 wherein the toggle means for a least-significant-bit (LSB) of the sum bits is an inverter that inverts the current state bit to generate the sum bit for a LSB bit-position;  
wherein the toggle means for bit-positions above the LSB bit-position each comprise an exclusive-OR (XOR) gate that has a second control input, the second control input being activated to cause the toggle means to invert the

current state bit, but being de-activated to cause the toggle means to not invert the current state bit.

[c15] 15.The sequencer of claim 14 wherein the sum bits and the current state bits include a first group, a second group, and a third group of bit-positions; wherein the combinatorial logic means further comprises a first logic grouping that generates sum bits in the first group, a second logic grouping that generates sum bits in the second group, and a third logic grouping that generates sum bits in the third group; wherein the first logic grouping of the combinatorial logic means does not receive the first and second pipelined carry signals but only receives current state bits from the first group; wherein the second logic grouping of the combinatorial logic means does not receive the second pipelined carry signal but only receives current state bits from the second group and receives the first pipelined carry signal; wherein the third logic grouping of the combinatorial logic means receives the first and second pipelined carry signals and receives current state bits from only the third group.

[c16] 16.The sequencer of claim 15 wherein a lowest bit-position in the second logic grouping has a toggle means that receives the current state bit for the bit-position and receives the first pipelined carry signal as the second control input; wherein other bit-positions in the second logic grouping have a toggle means with a second control input driven by a detect means for detecting when all lower bit-positions are in a pre-determined state that toggles the sum bit in the bit-position.

[c17] 17.The sequencer of claim 15 wherein each bit-position in the third logic grouping comprises:  
toggle means, receiving the current state bit for the bit-position, for toggling the current state bit to generate the sum bit in response to a second control input that is driven by a control signal for the bit-position; and  
detect means, driving the control signal to the toggle means, for detecting when all when all lower bit-positions are in a pre-determined state that toggles the

sum bit in the bit-position;

wherein a lowest bit-position in the third logic grouping has a detect means that receives the first and second pipelined carry signals;

wherein other bit-positions in the third logic grouping have a detect means that receives the first and second pipelined carry signals and at least one of the current state bits in the third group.

[c1 8] 18.The sequencer of claim 17 wherein the toggle means comprises an exclusive-OR (XOR) gate and wherein the detect means comprises an AND gate.

[c1 9] 19.A pipelined-carry incrementer comprising:  
a first state register receiving a first sum bit and outputting a first state bit synchronized to a clock;  
a second state register receiving a second sum bit and outputting a second state bit synchronized to the clock;  
a third state register receiving a third sum bit and outputting a third state bit synchronized to the clock;  
a fourth state register receiving a fourth sum bit and outputting a fourth state bit synchronized to the clock;  
a fifth state register receiving a fifth sum bit and outputting a fifth state bit synchronized to the clock;  
a sixth state register receiving a sixth sum bit and outputting a sixth state bit synchronized to the clock;  
a seventh state register receiving a seventh sum bit and outputting a seventh state bit synchronized to the clock;  
a first carry register receiving a first pre-carry and outputting a first pipelined carry synchronized to the clock;  
a second carry register receiving a second pre-carry and outputting a second pipelined carry synchronized to the clock;  
a first pre-carry gate, receiving the first, second, and third sum bits and outputting the first pre-carry;  
a second pre-carry gate, receiving the fourth and fifth sum bits and outputting the second pre-carry;  
second toggle logic that toggles the second state bit to generate the second

sum bit when the first state bit is high;  
third toggle logic that toggles the third state bit to generate the third sum bit when the first state bit and the second state bit are both high;  
fourth toggle logic that toggles the fourth state bit to generate the fourth sum bit when the first pipelined carry is high;  
fifth toggle logic that toggles the fifth state bit to generate the fifth sum bit when the first pipelined carry and the fourth state bit are both high;  
sixth toggle logic that toggles the sixth state bit to generate the sixth sum bit when the first pipelined carry and the second pipelined carry are both high;  
seventh toggle logic that toggles the seventh state bit to generate the seventh sum bit when the first pipelined carry and the second pipelined carry and the sixth state bit are all high.

[c20]

20. The pipelined-carry incrementer of claim 19 wherein the third, fifth, sixth, and seventh toggle logic comprise an AND gate that drives an input to an exclusive-OR (XOR) gate;  
wherein the second and fourth toggle logic comprise an XOR gate.